

# Capturas de Kotlin en Clase

## Tipos y Variables

```
fun main() {
    var patata = "Antes, todo esto, era campo">//variable
    var texto:String = "Le Text">//Variable definiendo tipo cadena
    var numerete:Int = 33//variable definiendo tipo Entero
    var doblete: Double = 65.7//Variable Double
    var flotador: Float = 67.6f//Float debe llevar una f al final
    val cuack = "pato">//Constante
    println("Hello, world!!!")
    println("$patata $texto $numerete $doblete $flotador $cuack")
}
```

```
Hello, world!!!
Antes, todo esto, era campo Le Text 33 65.7 67.6 pato
```

## Listas

```
fun main() {
    var listaCiclista: MutableList<String> = mutableListOf("Patata", "Zanahoria", "Claudio", "Pato") //Creamos lista y definimos contenidos de esta
    println(listaCiclista)//Imprimimos lista en pantalla
    println(listaCiclista[0])//Imprimimos en pantalla el primer elemento de la lista (El que sería el elemento 0)
    listaCiclista.add("Campo")//Añadimos elemento a la lista
    println(listaCiclista)
    listaCiclista.remove("Zanahoria")//eliminamos elementos de la lista
    println(listaCiclista)
    listaCiclista.count()//Cuenta el número de elementos de la lista
    println("Hay ${listaCiclista.count()}")//Mostramos en pantalla el número de elementos en la lista
}
```

```
[Patata, Zanahoria, Claudio, Pato]
Patata
[Patata, Zanahoria, Claudio, Pato, Campo]
[Patata, Claudio, Pato, Campo]
Hay 4
```

## Conjuntos

```
fun main() {
    var conjuntoCiclistas: Set<String> = setOf("nokia","Ex","LeQuique")
    //funciona Igual que las Listas mas o menos
}
```

# Mapas

Se pueden usar para almacenar datos de una forma similar a una base de datos

```
fun main() {
    //Map<Clave, ValorAlmacenado>
    val mapaCiclista: Map<String, Int> = mapOf("patata" to 1, "Zanahoria" to 2, "No se rick, parece falso" to 3)
    val mapaCiclista2: Map<Int, String> = mapOf(1 to "dani", 2 to "pepe", 3 to "cuacko")
    val mapaCiclista3: Map<Int, List<String>> = mapOf(1 to listOf("a","b","c","d"), 2 to listOf("d","e","f"))
    println(mapaCiclista)
    println(mapaCiclista2)
    println(mapaCiclista3)
    println(mapaCiclista2[2])//Ejemplo para acceder, por ejemplo, al valor pepe. Buscaríamos por clave
}
```

```
{patata=1, Zanahoria=2, No se rick, parece falso=3}
{1=dani, 2=pepe, 3=cuacko}
{1=[a, b, c, d], 2=[d, e, f]}
pepe
```

# Flujos de Control

## Bucle FOR

```
fun main() {
    val clasificacion: Map<Int, String> = mapOf(1 to "SeppKuss", 2 to "Kiam Auisp", 3 to "Mikel Landa")
    for(ciclista in clasificacion){//Bucle for para recorrer el mapa
        println(ciclista.value)
        println(ciclista.key)
    }

    for(i in 1..clasificacion.count()){//For desde 1 hasta la cantidad de elementos en el mapa
        println(i)
    }
}
```

```
SeppKuss
1
Kiam Auisp
2
Mikel Landa
3
1
2
3
```

## Sentencia IF

```
fun main() {  
    val clasificacion: Map<Int, String> = mapOf(1 to "SeppKuss", 2 to "Kiam Auisp", 3 to "Mikel Landa")  
    for(ciclista in clasificacion){//Bucle for para recorrer el mapa  
        if(ciclista.key == 1){//Condicional IF  
            println("El ciclista ${ciclista.value} ha ganado")  
        }else{  
            println("El ciclista ${ciclista.value} ha quedado en la posición ${ciclista.key}")  
        }  
        println(ciclista.value)  
        println(ciclista.key)  
    }  
  
    for(i in 1..clasificacion.count()){//For desde 1 hasta la cantidad de elementos en el mapa  
        println(i)  
    }  
}
```

```
El ciclista SeppKuss ha ganado  
SeppKuss  
1  
El ciclista Kiam Auisp ha quedado en la posición 2  
Kiam Auisp  
2  
El ciclista Mikel Landa ha quedado en la posición 3  
Mikel Landa  
3  
1
```

From:

<https://www.knoppia.net/> - Knoppia

Permanent link:

<https://www.knoppia.net/doku.php?id=kotlin:capturas&rev=1695221614>

Last update: **2023/09/20 14:53**

